

JJGO / dslogs

Last active last month • Report abuse

Star

[Code](#) [Revisions \(2\)](#) [Stars \(31\)](#) [Forks \(5\)](#)

Script for pretty printing logs of MIT 6.824 labs

[dslogs](#)

```

1 #!/usr/bin/env python
2 import sys
3 import shutil
4 from typing import Optional, List, Tuple, Dict
5
6 import typer
7 from rich import print
8 from rich.columns import Columns
9 from rich.console import Console
10 from rich.traceback import install
11
12 # fmt: off
13 # Mapping from topics to colors
14 TOPICS = {
15     "TIMR": "#9a9a99",
16     "VOTE": "#67a0b2",
17     "LEAD": "#d0b343",
18     "TERM": "#70c43f",
19     "LOG1": "#4878bc",
20     "LOG2": "#398280",
21     "CMIT": "#98719f",
22     "PERS": "#d08341",
23     "SNAP": "#fd971f",
24     "DROP": "#ff615c",
25     "CLNT": "#00813c",
26     "TEST": "#fe2c79",
27     "INFO": "#ffffff",
28     "WARN": "#d08341",
29     "ERRO": "#fe2626",
30     "TRCE": "#fe2626",
31 }
32 # fmt: on
33
34
35 def list_topics(value: Optional[str]):
36     if value is None:
37         return value
38     topics = value.split(",")
39     for topic in topics:
40         if topic not in TOPICS:
41             raise typer.BadParameter(f"topic {topic} not recognized")
42     return topics
43
44
45 def main(
46     file: typer.FileText = typer.Argument(None, help="File to read, stdin otherwise"),
47     colorize: bool = typer.Option(True, "--no-color"),
48     n_columns: Optional[int] = typer.Option(None, "--columns", "-c"),
49     ignore: Optional[str] = typer.Option(None, "--ignore", "-i", callback=list_topics),
50     just: Optional[str] = typer.Option(None, "--just", "-j", callback=list_topics),
51 ):

```

```
52     topics = list(TOPICS)
53
54     # We can take input from a stdin (pipes) or from a file
55     input_ = file if file else sys.stdin
56
57     # Print just some topics or exclude some topics (good for avoiding verbose ones)
58     if just:
59         topics = just
60     if ignore:
61         topics = [lvl for lvl in topics if lvl not in set(ignore)]
62
63     topics = set(topics)
64     console = Console()
65     width = console.size.width
66
67     panic = False
68     for line in input_:
69         try:
70             time, topic, *msg = line.strip().split(" ")
71             # To ignore some topics
72             if topic not in topics:
73                 continue
74
75             msg = " ".join(msg)
76
77             # Debug calls from the test suite aren't associated with
78             # any particular peer. Otherwise we can treat second column
79             # as peer id
80             if topic != "TEST":
81                 i = int(msg[1])
82
83             # Colorize output by using rich syntax when needed
84             if colorize and topic in TOPICS:
85                 color = TOPICS[topic]
86                 msg = f"[{color}]{msg}[/{color}]"
87
88             # Single column printing. Always the case for debug stmts in tests
89             if n_columns is None or topic == "TEST":
90                 print(time, msg)
91             # Multi column printing, timing is dropped to maximize horizontal
92             # space. Heavylifting is done through rich.column.Columns object
93             else:
94                 cols = ["" for _ in range(n_columns)]
95                 msg = "" + msg
96                 cols[i] = msg
97                 col_width = int(width / n_columns)
98                 cols = Columns(cols, width=col_width - 1, equal=True, expand=True)
99                 print(cols)
100            except:
101                # Code from tests or panics does not follow format
102                # so we print it as is
103                if line.startswith("panic"):
104                    panic = True
105                # Output from tests is usually important so add a
106                # horizontal line with hashes to make it more obvious
107                if not panic:
108                    print("#" * console.width)
109                    print(line, end="")
110
111    if __name__ == "__main__":
112        typer.run(main)
```